WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Wednesday, December 08, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR	
☐	L32	l12 and L30	5
☐	L31	l19 and L30	0
☐	L30	L28 same monitor$4	239
☐	L29	L28 same (information adj monitor)	0
☐	L28	L27 same controller	2478
☐	L27	(connect$4 adj processor)	16648
☐	L26	l17 and l12	7
☐	L25	l15 and l19	9
☐	L24	l12 and l15	45
☐	L23	l12 and l19	22
☐	L22	l19 same authenticat$4	4
☐	L21	l20 same monitor$4	3
☐	L20	L19 same determin$4	67
☐	L19	L18 same (server or (service adj provider))	278
☐	L18	(source near2 (terminal or node or computer)) same (destination near2 (terminal or node or computer))	4017
☐	L17	L16 same source same destination	43
☐	L16	L15 same (server or service)	229
☐	L15	(determin$4 near5 destination) same ((rout$4 or direct$4 or transmit$4 or transfer$4 or send$4 or forward$4) near3 request)	710
☐	L14	l5 and L12	31
☐	L13	l1 and L12	23
☐	L12	l8 or l9 or l10 or L11	5130
☐	L11	713/150,151,155,168,169.ccls.	1034
☐	L10	709/224,250.ccls.	2713
☐	L9	370/351.ccls.	574
☐	L8	710/36,38.ccls.	877
☐	L7	l5 same ((rout$4 or direct$4 or transmit$4 or transfer$4 or send$4 or forward$4) near3 request)	13
☐	L6	L4 same (monitor$4 near5 request)	7
☐	L5	L4 same monitor$4	200

☐	L4	(source same destination same network$4 same (service or server))	3087
☐	L3	L1.ab.	18
☐	L2	L1 with monitor$4	2
☐	L1	datalink adj layer	206

END OF SEARCH HISTORY

L2: Entry 1 of 2 File: USPT Jan 19, 1999

DOCUMENT-IDENTIFIER: US 5862335 A

** See image for Certificate of Correction **

TITLE: Method and apparatus for monitoring file transfers and logical connections in a computer network

CLAIMS:

13. A computer-implemented method of monitoring logical connections in a computer network, the computer network including a first multiplicity of stations, during a connection a pair of stations exchanging packets via the computer network, each packet including protocol control information and user data, the protocol control information including datalink layer information, network layer information, transport layer information, and a total number of bytes of user data associated with the packet, the datalink layer information including a source station identifier, a destination station identifier, and a network protocol identifier, the network layer information including a source network entity identifier and a destination entity identifier, and a type field, each packet including one of connection id and a pair of transport entity identifiers, the computer network including a connection record database having a second multiplicity of records, each record including identifiers for a pair of stations associated with an existing logical connection, one of a connection id and a pair of transport entity identifiers, a start of activity timestamp, a last activity timestamp, and a total number of bytes transferred during the logical connection, the method comprising the computer-implemented steps of:

a) determining whether a packet is part of a logical connection upon receipt of the packet;

b) ignoring the packet if it is not part of a logical connection;

c) searching the connection record database for a third multiplicity of records identifying a pair of stations matching the source station identifier and the destination station identifier;

d) creating a new record for the packet if no record is found with a pair of station identifiers matching the source station identifier and the destination station identifier of the packet;

e) searching the third multiplicity of records for a first set of records including a pair of network entity identifiers matching the source and destination network entity identifiers of the packet;

f) creating a new record for the packet if no record is found including a pair of network entity identifiers matching the source and destination network entity identifiers of the packet;

g) determining whether the packet uses a Novell protocol by examining the type field of the network layer information of the packet;

h) if the packet uses the Novell protocol:

1) searching the first set of records for a selected record, the selected record including a connection id matching the connection id of the packet,;

2) creating a new record for the packet if no record is found including a connection id matching the connection id of the packet;

i) if the packet does not use the Novell protocol:

1) searching the first set of records for a selected record, the selected record including a pair of transport entity identifiers matching a source and destination transport entity identifiers of the packet;

2) creating a new record for the packet if no record is found including a pair of transport entity identifiers matching the source and destination transport entity identifiers of the packet; and

j) updating the selected record by changing the last activity timestamp of the selected record to a value representative of a current time and updating the total number of bytes of user data transferred during the connection by the number of bytes of user data associated with the packet.

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L3: Entry 3 of 18 File: USPT Dec 28, 1999

DOCUMENT-IDENTIFIER: US 6009102 A
TITLE: NHRP packet authentication method and NHRP server

Abstract Text (1):
An authentication method in an NHRP (Next Hop Resolution Protocol) for performing an address resolution for converting a network layer address in an NBMA (Non-broadcast, Multi-access) network to a datalink layer address. The method comprises steps of: providing an NHRP server for performing an address resolution which has a plurality of interfaces belonging to respective sub-networks, maintaining authentication keys and authentication types respectively allocated to the interfaces in the NHRP server; authenticating an NHRP packet received from one of the interfaces by using the authentication key allocated to the interface which receives the NHRP packet; and discarding the NHRP packet in case of authentication being unauthorized. The method is capable of setting for each domain a mode for redirecting an NHRP (Next Hop Resolution Protocol) packet when authentication types between LIS (Logical IP Subnet, IP: Internet Protocol) are different and a mode for when an NHRP packet is redirected between domains.

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L6: Entry 2 of 7 File: USPT Jun 8, 2004

DOCUMENT-IDENTIFIER: US 6748431 B1
TITLE: Systems and methods for monitoring network exchanges between a client and a server

CLAIMS:

22. A networked computer monitoring system that allows a user to analyze the requests and responses to verify that the requests and responses occurred correctly and as expected, the monitoring system comprising: a client; a server, wherein the server receives a request from the client and in turn sends a response to the client; and a processor, wherein the processor performs the acts of: preserving a copy of all packets corresponding to the request and response in one or more buffers within one or more buckets that each correspond to a unique combination of source port and destination port for the client and the server such that one bucket is identified to preserve the copy of all packets for a given source port and destination port combination separately from one or more other packets from one or more other exchanges for one or more other source port and destination port combinations; coalescing data within the copy that is chunked; and flushing each buffer with a copy of all packets corresponding to the request and response.

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L6: Entry 3 of 7 File: USPT Jul 29, 2003

DOCUMENT-IDENTIFIER: US 6601082 B1
TITLE: System and method for managing actions provided by a network using a policy tree

Detailed Description Text (39):
An advantage of the present invention is that it provides for a comprehensive framework for the specification, configuration, administration and enforcement of policies related to the use of system or network actions/services such as a multicast access, a bandwidth allocation, a firewall security, etc. The present invention may allow a system and network administrator to exercise control over the providing of actions based on a particular source's attributes, time-of-day, group memberships, and source/destination networks/hosts, applications, etc. Generally stated, the present invention may provide a comprehensive management control over the use of network actions, i.e., to facilitate the specification, configuration, monitoring, and enforcement of policies related to requests for access to and reservation of actions in the network.

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L6: Entry 4 of 7 File: USPT May 27, 2003

DOCUMENT-IDENTIFIER: US 6570867 B1
TITLE: Routes and paths management

Detailed Description Text (37):
In order to service the user's historical route trace request, the queuing manager 60 periodically requests that the DB manager 46 provides it with a list of all source and destination object pairs stored in the route list of the database 25 and places them in its low priority queue. The queuing manager 60 will then sequentially send, in accordance with its defined priority levels, the low priority object pairs queued to the source and destination register 56 for processing and subsequent forwarding to the path assembler 57. The path objects assembler 57 operates in the same tanner it operates when assembling (discovering) paths in response to a real-time route trace request. Accordingly, in this case, the path objects assembler 57 operates to assemble all of the forward and backward paths associated with the route specified, requests that the path change monitor logic 59 monitor the paths discovered at the polling rate and notify the user via the notification channel 44 of any path changes observed as a result of a new poll or a trap received from the IP network 10.

Previous Doc Next Doc Go to Doc#

L6: Entry 6 of 7 File: USPT Feb 13, 2001

DOCUMENT-IDENTIFIER: US 6189043 B1
TITLE: Dynamic cache replication in a internet environment through routers and servers utilizing a reverse tree generation

Detailed Description Text (4):
FIG. 1B shows the next stage of the network organizing itself for optimizing the distribution of replica caches of information among the regions. Router 30, or a processor associated with router 30, transmits a monitor request message to all of the other routers 32, 34, 36, 38, and 39 connected to the backbone network 10 requesting the other routers to monitor service requests originating in their regions, that are directed to the primary cache in server 40. FIG. 2A shows an example of a monitor request message 200 that is sent out by Router 30 to all of the other routers 32, 34, 36, 38, and 39 connected to the backbone network 10. The monitor request message 200 includes a field 202 specifying the destination router, a field 204 specifying the source router, and a field 206 specifying the monitor request. The monitor request message 200 is broadcast to all other routers 32, 34, 36, 38, and 39 in the network.

Detailed Description Text (8):
The memory 302 in FIG. 3 stores the traffic monitoring program 329. It runs under the operating system program 320, with its instructions being executed on the processor 310. In routers 32, 34, 36, 38, and 39 which receive the monitor request message 200 over their respective network adapters 308, the traffic monitoring program 329 reads the message 200 buffered in buffer 326 and performs the traffic monitoring operation shown in FIG. 1C. In each of the destination routers 32, 34, 36, 38, and 39, the traffic monitoring program 329 monitors all messages that the router handles and counts the number of service requests directed to server 40 that request access to the data set stored as the primary cache "P". The traffic monitoring program 329 in the originating router 30 also performs this monitoring operation for the users 50 in the router's own region 20. The traffic monitoring program 329 in each of the routers 30, 32, 34, 36, 38, and 39 responds to router 30 with a traffic value that is the accumulated count of such service requests over a given period of time. The memory 302 in FIG. 3 stores the monitor response message buffer 328 that stores the monitor response message 220. The monitor response message 220 includes a field 222 specifying the destination router, a field 224 specifying the source router, and a field 226 specifying the monitor response which is the traffic value measured at that router. The monitor response message 220 is then transmitted by each router 32, 34, 36, 38, and 39 back to the originating router 30.

L22: Entry 1 of 4 File: USPT Dec 7, 2004

DOCUMENT-IDENTIFIER: US 6829232 B1
TITLE: Communication method for terminals connected to IP networks to receive services provided by intelligent networks

Detailed Description Text (45):
The server 3a which has received the admission request reads a program for executing the above-mentioned processing flow of FIG. 5 from a memory and executes the program. First, an authentication process is performed by using the identifier of the source terminal included in the received signal. The identifier of the source terminal included in the received signal is compared with destination number information to check whether the call set between them is a call in the same area or not. When the call is not made in the same area, the class of the destination number is checked by using the information prestored in the memory 42. In the case of the class in which the IP address of the server 3b controlling the gateway to which the destination terminal is connected is unconditionally determined from the destination number, the IP address of the server 3b and the information (such as port number of the TCP) used to transmit/receive the call process signal information are retrieved (202).

Detailed Description Text (57):
First, an authentication process is performed by using the identifier of the source terminal included in the admission request from the server 3a. The identifier of the source terminal included in the received signal is compared with destination number information to check whether the call is made in the same area or not. When the call is not in the same area, the class of the destination number is checked by using the information prestored in the memory 42. In the case of the class in which the IP address of the server for controlling the gateway to which the destination terminal is connected is not unconditionally determined from the destination number, the information in the cache in the memory of the server 3a is retrieved. When the information related to the destination number is not included in the cache, an interrogation request is multicasted to a plurality of servers with the function of accessing the service control gateways 1 (1a, 1b) (252, 253).

L22: Entry 2 of 4 File: USPT Sep 28, 2004

DOCUMENT-IDENTIFIER: US 6798782 B1
TITLE: Truly anonymous communications using supernets, with the provision of topology hiding

Detailed Description Text (26):
FIG. 7 depicts a flow chart of the steps performed when sending a packet from node A. Although the steps of the flow chart are described in a particular order, one skilled in the art will appreciate that these steps may be performed in a different order. Additionally, although the SNSL layer is described as performing both authentication and encryption, this processing is policy driven such that either authentication, encryption, both, or neither may be performed. The first step performed is for the SNSL layer to receive a packet originating from node A via the TCP/UDP layer and the inner IP layer (step 702). The packet contains a source node ID, a destination node ID, and data. The SNSL layer then accesses the VARPDB to obtain the address mapping between the source node ID and the source real address as well as the destination node ID and the destination real address (step 704). If they are not contained in the VARPDB because this is the first time a packet has been sent from this node or sent to this destination, the VARPDB accesses the local VARPD to obtain the mapping. When contacted, the VARPD on the local machine contacts the VARPD that acts as the server for the Supernet to obtain the appropriate address mapping.

L22: Entry 3 of 4 File: USPT Jul 16, 2002

DOCUMENT-IDENTIFIER: US 6421536 B1
TITLE: Communication system

Detailed Description Text (141):
Further, for example, an operation can easily be realized that by using transfer additional information, the transfer destination computer 1504 acquires information from the transfer source computer 1505 via the information server 1506 and the information server 1506 performs authentication for such information acquisition. This will be explained below for a case where the transfer source computer 1505 stores information in the information server 1506 before sending a holding transfer request 1802. In this case, an information identifier generating means is provided in the information server 1506. When the transfer source computer 1505 has stored information in the information server 1506, the information identifier generating means generates an information identifier, which is sent to the transfer source computer 1505. The transfer source computer 1505 causes the address of the information server 1506 and the information identifier generated by the information server 1506 to be accommodated in a holding transfer request 1802 as transfer additional information. As a result, the transfer destination computer 1504 that has acquired the transfer additional information can send, by using the address of the information server 1506, a request for acquiring information from the information server 1506. If the transfer destination computer 1504 also sends the information identifier at this time, the information server 1506 can not only judge, by collating the information identifier, whether it is proper to send information to the transfer destination computer 1504 but also determine the information to be sent. For example, the use of such an information identifier enables a control that even in a case where usually the transfer destination computer 1504 is not permitted to acquire information from the information server 1506, the transfer destination computer 1504 can acquire only information that is associated with an information identifier from the information server 1506.

L26: Entry 1 of 7 File: USPT Nov 30, 2004

DOCUMENT-IDENTIFIER: US 6826606 B2
TITLE: Method and apparatus for communicating among a network of servers

Detailed Description Text (238):
The request event 700 passes to the event delivery object 312 of the event bus 310. Assume that the service locator 354 determines the target server to be the remote server 180'. The event delivery object 312 then determines (step 912) from the destination host parameter of the SendEventandWait command that the destination subsystem 300' is on the remote server 180'. because the destination subsystem 300' is remote to the source subsystem 300, the request event 700 passes (step 914) to the transport layer 318 on the server 180. The transport layer 318 then transmits (step 916) the request event over the network connection 200 to the transport layer 318' on the server 180'.

Current US Cross Reference Classification (2):
709/224

L26: Entry 6 of 7 File: USPT Oct 16, 2001

DOCUMENT-IDENTIFIER: US 6304912 B1
TITLE: Process and apparatus for speeding-up layer-2 and layer-3 routing, and for determining layer-2 reachability, through a plurality of subnetworks

Brief Summary Text (15):
According to the NHRP, a server called next hop server (NHS) is provided for each LIS, and each next hop server usually functions as a router between LISs. A terminal (or station) which establishes a virtual connection in accordance with the NHRP protocol is called next hop client (NHC). Each next hop client registers its own IP address and ATM address in the next hop server of the LIS to which the next hop client belongs. When a next hop client which wants to establish a virtual connection to a destination host, the next hop client, which is denoted here as a source next hop client, sends an NHRP request packet to the next hop server in the LIS to which the source next hop client belongs, where the NHRP request packet contains an IP address of the destination host. When the next hop server receives the NHRP request, and determines that the destination host belongs to an LIS which is different from the LIS which the next hop server controls, the next hop server transfers the NHRP request to another NHS (next hop server) in accordance with its own routing table and the IP address of the destination host. Thus, the NHRP request packet is transferred through a hop-by-hop path passing through routers until the NHRP request packet reaches a next hop server to which controls the LIS which the destination host belongs. When a next hop server, which receives the NHRP request, determines that the destination host belongs to the LIS which the next hop server controls, the next hop server searches its own ATM address table for the ATM address corresponding to the IP address of the destination host, and sends an NHRP response containing the ATM address of the destination host, to the source next hop client which has sent the NHRP request, tracing back the path through which the above the NHRP request has been transferred. When the next hop client receives the NHRP response containing the ATM address, the next hop client establishes a shortcut virtual connection (shortcut VC) to the destination host based on the received ATM address. The shortcut virtual connection does not pass a router, and therefore is free from the delays due to packet reconstruction and transfer in the router, to achieve high speed communication. Each next hop client communicates to another host in an LIS which is different from the LIS of the next hop client, through the hop-by-hop path passing through routers, until the ATM address is obtained as above, and then the hop-by-hop path is switched to the above shortcut virtual connection after the ATM address is obtained and the virtual connection established.

Current US Cross Reference Classification (1):
370/351

L25: Entry 2 of 9 File: USPT Sep 2, 2003

DOCUMENT-IDENTIFIER: US 6614796 B1
TITLE: Fibre channel arbitrated loop bufferless switch circuitry to increase bandwidth without significant increase in cost

Brief Summary Text (2):
Fibre Channel networks are known loop configuration networks that have a plurality of known type nodes such as servers, printers, disk arrays etc. all connected together by the loop. Such networks use a unique protocol involving a plurality of 40 bit primitives that are used to arbitrate for loop control, to establish connections and to carry out flow control for data transfers of frames of data. The flow control inherent to the Fibre Channel Arbitrated Loop network (hereafter FCAL nets) protocol has the advantage of eliminating the need for the nodes to have extensive buffering capabilities since the destination node controls the amount of data it receives by transmission of an RRDY primitive to the source node each time the destination node is ready to: receive another frame.

Detailed Description Text (37):
There are several alternative embodiments for establishing the back channel simplex connection through the backplane. One involves updating all the scoreboards of all ports with information as to which source node address is coupled to any port which is indicated in the scoreboard as having a busy status and is thus available for dual simplex. In this alternative embodiment, the source port posts a message to the protocol bus to update all scoreboards in every port to indicate that although it is busy, it is coupled to the source node of the loop tenancy and is thus available to receive data in a dual simplex mode. In this alternative embodiment, the third port checks its scoreboard, and if it determines that its destination node has the same address as the source node coupled to the source port, it then sends a connection request message. If it determines from the scoreboard data that the source port is not dual simplex capable, it does not send a connection request message.

Detailed Description Text (92):
After the OPN and any following RRDYs from the source node are latched, the port that latched the OPN determines the port ID of the port coupled to the destination node by consulting the routing table as described above. After determining the ID of the remote destination port, that port ID is used as a search key to consult the scoreboard table to determine if the port is available and if there is the privilege to talk to it. If the remote ports status is "busy camp" and camping is allowed, send a connect request message and wait for the remote port to finish its current conversation, grant the connect request and send back a connect response message naming the backplane channel to use. If the remote port's status is "no privilege", generate an OPN and send it back to the source node. If the remote port's status is "available", send a connect request message to the remote port over the protocol bus. If the status if "busy no camp", return a CLS to the source node. This causes the remote port to arbitrate for and win control of its local loop. When control is won, the remote port then sends a reply message to update its scoreboard status to busy and naming the backplane channel to use. The scoreboard circuitry sees the reply message and updates the status of the port whose ID is in the reply message to busy. The reply message causes the first port to generate signals to the crossbar switch to open a connection between the two ports. The

L25: Entry 6 of 9 File: USPT Aug 28, 2001

DOCUMENT-IDENTIFIER: US 6282563 B1
TITLE: Agent moving method, message transmission method, agent transmission method, agent acquisition method, message acquisition method and agent list acquisition method, and computer

Drawing Description Text (15):
FIG. 13 is a diagram showing formats for data that are exchanged between a source computer, a destination computer and an agent temporary storage server in an agent moving arrangement where a temporary storage area is provided.

Drawing Description Text (16):
FIG. 14 is a diagram showing formats for data that are exchanged between a source computer, a destination computer and an temporary storage server in a message transfer arrangement where a temporary storage area is provided.

Detailed Description Text (31):
FIG. 5 shows an example arrangement for the movement of an agent when a temporary storage area is provided. The agent source computer 3 includes a logical space 6, where the agent is active, and a transmitter 7 of an agent movement processing mechanism. The agent 1 is active in the logical space 6. The transmitter 7 of the agent movement processing mechanism comprises an agent freezing unit 22, a moving method determination unit 23, an agent direct movement request transmitter 24, and an agent temporary storage request transmitter 25. The agent freezing unit 22 performs an agent freezing process, and the moving method determination unit 23 performs a process for determining whether a mobile agent can move directly to the agent destination computer 4. There are many references available for determining whether an agent can move directly to a destination computer or not. For example, an agent can not move directly to a destination computer when the computer is not powered on or when direct access from an external network is inhibited. The agent direct movement request transmitter 24 must transmit a transfer request 34 of the bit sequence 11 for the frozen agent to the agent destination computer 4. In other words, the agent direct movement request transmitter 24 has address information of the agent destination computer 4. The agent temporary storage request transmitter 25 transmits a storage request 35 of the bit sequence 11 of the frozen agent to an agent temporary storage server 21. In other words, the agent temporary storage request transmitter 25 correspondingly stores an address of the temporary storage server 21 for the agent destination computer 4.

Detailed Description Text (32):
The agent destination computer 4 comprises a logical space 6 where an agent is active, and a receiver 8 of an agent movement processing mechanism. The receiver 8 of the agent movement processing mechanism includes an agent reproduction unit 26, a transfer method confirmation unit 27, an agent direct movement request receiver 28, and a temporary agent storage acquisition request processing unit 29. The agent reproduction unit 26 reproduces an agent 1 from a bit sequence 11 for a frozen agent, and activates it in the logical space 6. The transfer method confirmation unit 27 receives a confirmation request 33 from the moving method determination unit 23 in the agent source computer 3, and transmits a response representing whether or not direct movement is possible. The agent direct movement request receiver 28 receives the bit sequence 11 for the frozen agent that is transmitted

from the agent source computer 3, and calls the agent reproduction unit 26. The temporary agent storage acquisition request processing unit 29 transmits an agent acquisition request 36 to the agent temporary storage server 21, receives the bit sequence 11 for the frozen agent that is transferred from a temporary agent storage processor 31, and calls the agent reproduction unit 26.

Detailed Description Text (35):
An example arrangement for a message transfer when a temporary storage area is provided is shown in FIG. 6. A message transfer source computer 13 comprises a logical space 6 where an agent is active and a transmitter 15 of a message transfer processing mechanism. An agent 1 is active in the logical space 6. In FIG. 6, the agent 1 transfers message 2 to an agent 1c that is active in a message transfer destination computer 14. The transmitter 15 of the message transfer processing mechanism includes a message/bit sequence converter 38 for a message, a transfer method determination unit 39, a message direct transfer unit 40, and a message temporary storage request transmitter 41. The message/bit sequence converter 38 converts the message 2 into a bit sequence 18 for the message. The transfer method determination unit 39 determines whether or not a message should be transferred directly to the message transfer destination computer 14. There are many references for determining whether or not the message can be directly transferred to the destination computer. When, for example, the destination computer is not powered on, when direct access from an external network is inhibited, or when a destination agent is stored in the temporary storage area and has not arrived at the destination computer, the direct message transfer is inhibited. In other cases, the direct message transfer is permitted. The message direct transfer unit 40 transfers a message direct transfer request 51 to the message transfer destination computer 14. In other words, the message direct transfer unit 40 possesses information concerning the address of the message transfer destination computer 14. The message temporary storage request transmitter 41 transfers a message temporary storage request 52 to the agent temporary storage server 21. That is, the message temporary storage request transmitter 41 correspondingly possesses the address of the temporary storage server 21 for the message transfer destination computer 14.

Detailed Description Text (36):
The message transfer destination computer 14 includes a logical space 6 where an agent is active and a receiver 16 of a message transfer processing mechanism. An agent 1c, which is a message transfer destination agent, is active in the logical space 6. The receiver 16 of a message transfer processing mechanism includes a message distribution unit 42, a transfer method confirmation unit 43, a message direct receiver 44, a temporarily stored message acquisition request processing unit 45 and a bit sequence/message converter 46. The message distribution unit 42 transmits a received message to the destination agent 1c. The transfer method confirmation unit 43 transmits a response relative to a confirmation request 50, which is issued by the transfer method confirmation determination unit 39 in the message transfer source computer 13, to confirm a direct transfer. The message direct receiver 44 receives a message direct transfer request 51 issued by the message direct transfer unit 40 in the message transfer source computer 13. The temporarily stored message acquisition request processing unit 45 transmits a temporarily stored message acquisition request 53 to the agent temporary storage server 21, and receives a temporarily stored message transfer request 54, which is returned as the result. The bit sequence/message converter 46 reproduces a message from the bit sequence 18 for the message received by the message direct receiver 44 and the temporarily stored message acquisition request processing unit 45.

Detailed Description Text (39):
The procedures employed for the agent movement method are shown in FIGS. 7, 8 and 9. FIG. 7 shows the processing performed when an agent that is active in the agent source computer 3 can be moved directly to the agent destination computer 4. FIG. 8 shows processing performed when the agent that is active in the agent source computer 3 can not be moved directly to the agent destination computer 4 and is

stored temporarily. FIG. 9 shows processing performed when the agent destination computer 4 acquires an agent that was stored in the agent temporary storage server 21.

Detailed Description Text (42):
FIG. 8 is a diagram showing the processing performed when the agent 1 that is active in the logical space 6 of the source computer 3 can not move directly to the logical space 6 in the destination computer 4, and is stored in the agent temporary storage server 21. In the moving method determination process 57, a procedure performed until a direct movement confirmation request 33a (FIG. 13(a)) is transmitted to the destination computer 4 is the same as that explained referring to FIG. 7. In FIG. 8, however, it is assumed either that the destination computer 4 has not been activated, or that a communication path from the source computer 3 to the destination computer 4 has been disconnected. In these cases, transmission of the direct movement confirmation request 33a in the moving method determination process 57 has failed. As a consequence of the failure, it is determined that an agent temporary storage process must be performed, and the agent temporary storage request transmitter 25 is called. The agent temporary storage request transmitter 25 performs an agent temporary storage request transmission process 63. An agent temporary storage request 35a (FIG. 13(e)) is transmitted to the agent temporary storage unit 30 in the agent temporary storage server 21 and the agent temporary storage request transmission process 63 waits for a response.

Detailed Description Text (43):
The agent temporary storage unit 30 in the agent temporary storage server 21 receives the agent temporary storage request 35a, and performs an agent temporary storage process 64. In this process, the bit sequence 11 of the frozen agent is extracted from the received request, and is stored at a location in the agent temporary storage area 32 allocated for the agent destination computer 4. Following this, an agent temporary storage confirmation response 35b (FIG. 13(f)) is transmitted to the agent temporary storage request transmitter 25 that issued the request. During an agent temporary storage request transmission process 63 performed by the source computer 3, the agent temporary storage confirmation response 35b is received, and the agent movement process is thereafter terminated.

Detailed Description Text (48):
FIG. 10 is a diagram showing the processing by which an agent 1 that is active in a message transfer source computer 13 directly transfers a message 2 to an agent 1c that is active in a message transfer destination computer 14. The agent 1, which is active in the logical space 6 in the message transfer source computer 13, generates the message 2 and begins transmission of the message. Then, a message/bit sequence converter 38 performs a message/bit sequence conversion process 70 to obtain a bit sequence 18 for the message 2. Then, the transfer method determination unit 39 performs a transfer method determination process 71. During this process, a direct transfer request 50a (FIG. 14(a)) is transmitted to the transfer method confirmation unit 43 in the message transfer destination computer 14, and a response is employed to determine the transfer method.

Detailed Description Text (51):
FIG. 11 is a diagram showing the processing that is performed when the agent 1 that is active in the message transfer source computer 13 can not directly transfer the message 2 to the agent 1c that is active in the message transfer destination computer 14, and the message 2 is temporarily stored in the agent temporary storage server 21. The processing performed until the transfer method determination unit 39 in the message transfer source computer 13 transmits the direct transfer request 50a is the same as that explained referring to FIG. 10. In the processing in FIG. 11, however, it is assumed that the message transfer destination computer 14 has not been activated, that a communication path from the message transfer source computer 13 to the message transfer destination computer 14 has been disconnected, or that a destination agent has not arrived. In the transfer method determination

process 71 performed by the transfer method determination unit 39, the transmission of the direct transfer confirmation request 50a fails if the destination computer has not been activated or if the communication path is interrupted. As a result, the transfer method determination unit 39 in the source computer 13 determines that the message should be stored in the agent temporary storage server 21. The same determination is made when the direct transfer rejection response 50c (FIG. 14(c)) is returned. As a result, the message temporary storage request transmitter 41 is called, and it performs a message temporary storage request transmission process 78. For this process 78, a message temporary storage request 52a (FIG. 14(f)) is transmitted to the message temporary storage unit 47 in the agent temporary storage server 21. The process 78 waits for a response from the agent temporary storage server 21.

Detailed Description Text (56):
The present invention is not limited to the above described embodiment. For example, an agent and message transfer destination computer may transmit a data acquisition request, and in response to this, a temporary storage server may add marks to agents and messages stored in an area for the destination computer, and transfer the marked agents and messages to the destination computer. At this time, when an agent has first been transmitted, and after the termination of the transmission, a message may be transmitted, or an agent and a message related to the agent may be transmitted sequentially. The determination of the source computer whether or not direct movement (transmission) of an agent is possible may be performed by the agent.

Detailed Description Text (58):
Another embodiment is also provided. FIG. 19 is a diagram showing another embodiment when an agent destination computer 4 acquires a bit sequence 11 for an agent and/or a bit sequence 18 for a message from an agent temporary storage server 21. There is a case where a user of the destination computer 4 wants to select an agent stored in the temporary storage server 21 and to execute only the selected agent at the destination computer 4. The user instructs a temporarily stored agent acquisition request unit 29 (FIG. 5) to begin a temporarily stored agent list acquisition process 87. In the temporarily stored agent list acquisition process 87, a temporarily stored agent list acquisition request 90a (FIG. 20(a)) is transmitted to the temporary storage server 21. Upon reception of the request 90a, a temporarily stored agent acquisition unit 31 (FIG. 5) in the temporary storage server 21 begins a temporarily stored agent list acquisition process 84. In the list acquisition process 84, the names of agents are acquired that are stored in the storage area for a source computer of the request 90a when the list acquisition request 90a has received, and a list is prepared. When the bit sequence 11 for an agent is stored in a file system shown in FIG. 15, the file name of the bit sequence 11 for the agent is acquired in the directory of the source computer. While an agent list is referenced in this embodiment, a message list can be handled in the same manner.

Previous Doc Next Doc Go to Doc#